Klasterovanje sekvence pogleda

Filip Jakovljević
Fakultet tehničkih nauka
Univerzitet u Novom Sadu
Trg Dositeja Obradovića 6
21000 Novi Sad
filipjakovljevic@uns.ac.rs

1 Uvod:

Čulo vida je primarno čulo ljudi preko kojih oni primaju informacije iz sveta. No pored toga što čulo vida primarno služi da prima informacije, ono može i da odaje informacije posmatraču sa strane. Činjenica je da su ljudi sposobni da primete ako čovek gleda u nešto sto mu se sviđa na osnovu načina kako čovek posmatra to nešto i njegovih izraza lica [1]. Danas se sve više istraživanja bavi ljudskim pogledom, od toga kako oči reaguju na različite boje, kako ljudi reaguju na pokrete u svom vidnom polju pa i do informacija koje se zanemaruju iz našeg vidnog polja.

Ovaj rad pokušava da uoči šablone pogleda, ako postoje prilikom rešavanja testova na računaru, praćenjem informacija o regionima interesa koje ljudi posmatraju prilikom njihovog rešavanja, i njihovoj skoncentrisanosti na neki od datih regiona. Ovim se dobija uvid u važnost regiona i njihovu korisnost za rešavanje problema. Pretpostavka je da postoje određeni šabloni u koje se mogu grupisati načini gledanja subjekta i da postoje grupe ljudi koje koriste isti šablon. Ovakve analize informacija mogle bi da budu korisne u raznim sferama ljudskih delatnosti gde postoji interakcija računara sa ljudima, ta interakcija bi mogla da se poboljša, unapredi i učini lagodnijom.

Tehnike mašinskog učenja su upotrebljene u svrhu rešavanja cilja pronalaženje sličnih grupa. Korišćene tehnike obuhvataju auto-enkoder kao neuronske mreže koje su zadužene za kompresiju podataka i izdvajanje bitnih informacija iz sekvenci pogleda kao i algoritmi za klasterizovanje podataka koji služe za grupisanje srodnih podataka u zasebne celine.

Dalja poglavlja obuhvataju srodna istraživanja gde će se dati kratak uvid projekte koje se bave ovom tematikom i tehnikama, model daje kratak opis korišćenog modela neuronske mreže, rezultati gde su prikazani dobije rezultati i opisan eksperiment i zaključak.

2. Srodna Istraživanja:

2.1 RNN i LSTM RNN mreža

RNN (Recurrent neural networks) su rekurentne neuronske mreže gde ćelije pored ulaza iz spoljašnje sredine prima ulaz i od same sebe. Efektivno izlaz ćelije u prethodnom koraku postaje deo ulaza te iste ćelije u narednom koraku. Rekurentne mreže u principu koriste njihovu povratnu vezu da čuvaju reprezentacije prethodnih ulaza i time na neki način omogućavaju memoriju [2]. Ovo je značajno za mnoge procese gde je potrebno pamćenje prethodnih stanja, i našle su primenu u obradi signala, sekvenci i vremenskih serija [4].

Problem koji se javlja prilikom korišćenja rekurentnih neuronski mreža jeste taj što one u teoriji mogu da pamte događaje iz proizvoljno duge prošlosti ali u praksi ta memorija je relativno kratka (Short Term Memory). Kako postoje aplikacije gde je prošlost podjednako bitna kao i sadašnjost stvorila se potreba za mrežama koje mogu da prevaziđu ovaj problem [3].

LSTM (Long Short Term Memory) su rekurentne neuronske mreže koje su razvijene da

prevaziđu nedostatak kratke memorije [2]. LSTM uvode novu arhitekturu ćelije, gde je se na postojeću ćeliju dodaju kapije koje treba da zaštite memoriju.

Kako u radu koristimo podatke koji su u sekvencama rekurentne neuronske mreže su prirodan izbor, i kako nam je podjednako bitno šta se dešavalo na početku sekvence kao i na kraju, odnosno treba nam dugoročna memorija LSTM ćelije se nameću kao potrebne.

2.2 Autoenkoderi

Autoenkoder (Autoencoder) je tip neuronske mreže koji je dizajniran da nauči skrivenu reprezentaciju podataka [8]. To postiže tako što pokušava da izvuče skrivenu reprezentaciju i na osnovu nje rekonstruiše ulaz, uči tako što minimizuje grešku rekonstrukcije [5]. Može se koristiti za mnoge stvari među kojima je i redukcija dimenzionalnosti [6] i nenadgledano učenje [7].

Sastoji se iz dva dela. Prvi deo zove se enkoder i zadužen je za učenje skrivene reprezentacije, dok je drugi deo zadužen za rekonstrukciju te reprezentacije u prvobitan ulaz i zove se dekoder. Enkoder i dekoder su obično simetrične neuronske mreže. Kada se obuči ceo enkoder ova dva dela mogu da se otkače i koriste zasebno, odnosno obično se enkoder deo sačuva i posle se na njegov izlaz zakači nova neuronska mreža koja služi za klasifikaciju.

Ovaj rad koristi autoenkoder kako bi naučio skrivene osobine i odbacio nebitne podatke u sekvenci, time efektivno redukova dimenzionalnost polazne sekvence, izlaz iz dela enkodera koristi se za klasterizaciju.

3. Model:

Za potrebe rada konstruisan je LSTM RNN Autoenkoder. Koji je sequence-to-sequence neuronska mreža, gde je cilj da se ulazna sekvenca prvo enkodira u skraćenu reprezentaciju (koja će se kasnije koristiti za klasterovanje) i zatim dekodira u izlaznu sekvencu koja treba da bude što sličnija ulaznoj sekvenci.

Arhitektura modela:

1: Input: 30x5

2: LSTM: 512, tanh, return sequences

3: LSTM: 256, tanh, return sequences

4: LSTM: 8, tanh

5: RepeatVector: 30

6: LSTM: 256, tanh, return sequences

7: LSTM: 512, tanh, return sequences

8: TimeDistributed-Dense: 5

4. Rezultati:

4.1 Postavka eksperimenta i prikupljanje podataka

Početni korak u rešavanju problema klasterovanja sekvenci gledanja ljudi prilikom rešavanja testova na računaru jeste prikupljanje podataka. Prvo se kod pitanja pripremljenih za testove obeležavaju regije od interesa koje mogu biti na primer: tekst, slika u pitanju, ponuđeni odgovori u pitanju itd. Ove regije od interesa su pravougaonog oblika i zapisuju se pomoću koordinata suprotnih temena pravougaonika. Pitanja u testu se prikazuju jedno po jedno i staju na ekran bez potrebe za skrolovanjem.

Mašina za praćenje pogleda nalazi se ispod monitora računara na kojem se radi test da bi skupljala podatke o tački u koju subjekat gleda. Prilikom pokretanja testa vrši se prvo kalibracija mašine, gde subjekat koji radi test gleda u zadate tačke na ekranu. Posle kalibracije se kreće sa skupljanjem podataka i prikazivanjem pitanja. Podaci koji se prikupe tokom rađenja testa se čuvaju na računaru za kasnije procesiranje. Podaci se sastoje iz procenata širine ekrana (X koordinate) tačke u koju subjekat gleda, procenta visine ekrana (Y koordinate) tačke u koju subjekat gleda i vremena.

4.2 Procesiranje podataka

Prvi korak u procesiranju podataka jeste pretvaranje koordinata zadatih u procentima u koordinate u piksela ekrana. Dalje se te koordinate mapiraju na regije od interesa u pitanju, svaka regija je obeležena sa jedinstvenim brojem, posle ove transformacije dobija se sekvenca regiona od interesa. Kako nigde nije dato vremensko ograničenje gledanja jednog pitanja, ove sekvence mogu biti proizvoljno dugačke veličine, te je potrebno njihovo skaliranje na određenu dužinu sekvence. Sekvence se skaliraju na dužinu od trideset regiona od interesa, gde se zadržavaju vremenski odnosi koliko je dugo koji region gledan (region koji je najduže gledan biće najduži i kad se skalira) i redosled smenjivanja regiona. Samo skaliranje će dovesti i do otklanjanja šumova u podacima, pošto će suviše kratki vremenski intervali na nekom regionu otpasti.

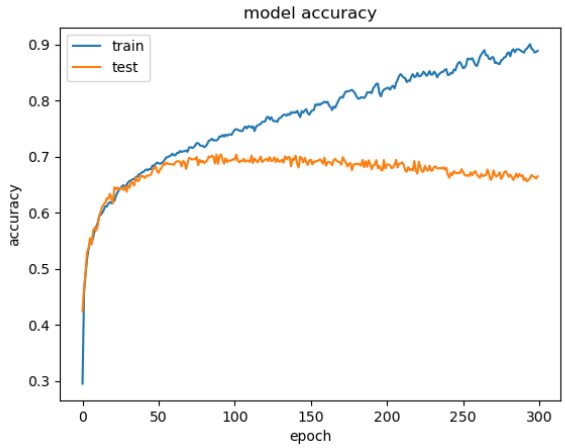
4.3 Obučavanje autoenkodera

Za obučavanje autoenkodera prikupljeno je 751 sekvenca pogleda, gde svaka sekvenca ima dužinu od 30 regiona. Kako se regioni od interesa označavaju prirodnim brojevima između njih postoji određena metrika udaljenosti koju mi nismo zadali, zbog toga se pre njihovog propuštanja u autoenkoder oznake regiona transformišu u one-hot vektor. Konačan ulaz u autoenkoder je 751x30x5 pošto postoji 5 regija od interesa na prikazanim testovima.

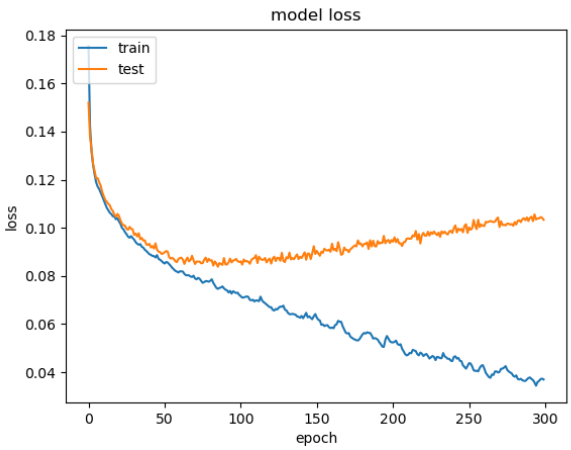
Testirane je više arhitektura autoenkodera i kao najbolja izabran je autoenkoder opisan u poglavlju model. To on sadrži u sebi dekoder koji iz sekvence od 30 regiona izvlači vektor osobina dužine 8. Skup podataka je podeljen na trening i test skup, gde je test skup 10% od celog skupa prikupljenih podataka. Autoenkoder je obučavan na 300 epoha i zadržan je samo model koji je ostvario najbolje rezultate.

Ispod slede grafici tačnosti i gubitka nastali prilikom obučavanja autoenkodera:

Slika 1. Grafik tačnosti kroz epohe



Slika 2. Grafik gubitka kroz epohe



Sačuvane su težine iz 86. epohe kada je model ostvario tačnost od 70,17% na test skupu podataka. Posle te epohe model počinje da se overfituje na podatke u obučavajućem skupu.

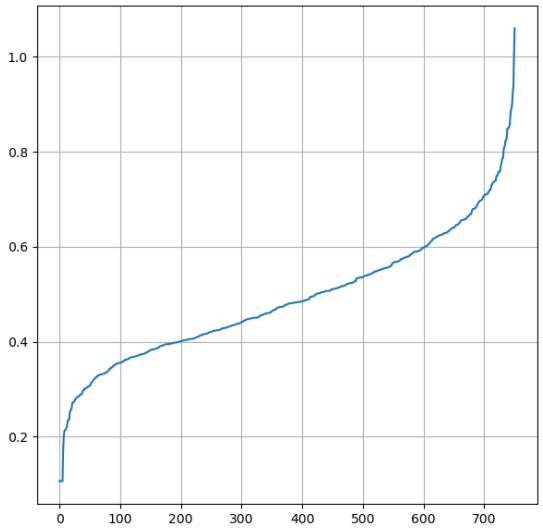
4.4 Klasterizacija

4.4.1 DBSCAN klasterizacija

Klasterizacija se vrši na izlazu enkoder dela i autoenkodera. Klasterizovani su vektori osobina dužine 8. Prvi proban metod klasterizacije je bio DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Metod koji na osnovu gustine i jezgro tačaka određuje pripadnost klasteru. Da bi se zadao

parametar epsilon prvo je iscrtan grafik pete najbliže tačke svake tačke skupa:

Slika 3. Udaljenost pete najbliže tačke od svake tačke



Kao epsilon izabrano je 0.6, a kao minimalan broj tačaka da bi neka tačka bila jezgro korišćeno je 5 tačaka. Pored ove kombinacije probane su i druge ali ni jedna od kombinacija nije pokazala dobre rezultate, uglavnom je velika većina tačaka pripadala samo jednom klasteru ili je pripadala šumu, a drugi klasteri su sadržali svega par tačaka.

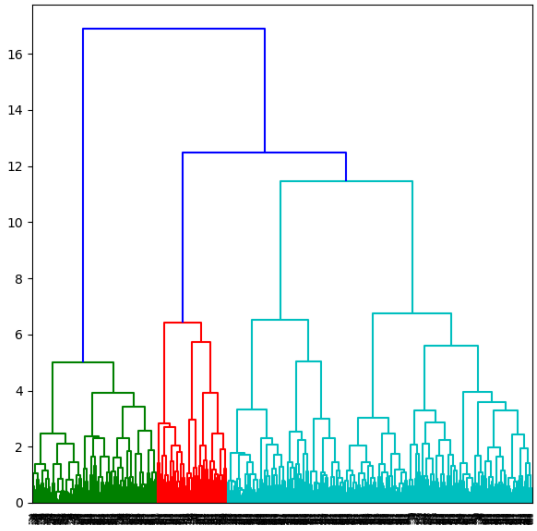
4.4.2 Hijerarhijska klasterizacija

U radu je korišćeno aglomerativno klasterovanje, gde je svaka tačka na početku klaster i u svakom narednom koraku se spajaju najbliži parovi tačaka dok ne preostane samo jedan klaster. Ovaj proces se može vizualizovati pomoću dendograma (dijagram u obliku stabla koji beleži sekvence spajanja ili razdvajanja).

Za hijerarhijsko klasterovanje nije potrebno znati broj klastera već se posmatra odgovarajući dendogram i na njemu se odluči gde se odseca za bilo koji željeni broj klastera.

Dendogram je napravljen od podataka koje je generisao enkoder:

Slika 4. Dendogram dobije od izlaza enkodera



Posmatrajući dendogram odlučeno je da se odseče na 4 klastera. I daljom analizom sekvenci koje su upale u te klastere može se zaključiti da pripadnost nekom klasteru najviše odredjuje najduže gledani region.

Prvi klaster čine sekvence gde se najduže posmatrao 0. region, slede primeri 5 nasumičnih sekvenci iz prvog klastera:

1. 00000000020000043333403004332
2. 00000000000442000000000034444
3. 00000000000000000000411000002
4. 32100000000000000000023233323
5. 41001000000004323000430000003 2

Drugi klaster čine sekvence gde se najduže posmatrao 4. region, slede primeri 5 nasumičnih sekvenci iz drugog klastera:

1. 000032224444444444444444444444
2. 011444444444444444444444444444
3. 000113404444444444444444444444
4. 200004444444444444444444444444
5. 021002104444334044444444444444

Treći klaster čine sekvence gde se najduže posmatrao 1. region, slede primeri 5 nasumičnih sekvenci iz trećeg klastera:

1. 041111111111144444444433333000
2. 000111110010011111111114211000
3. 112211111112211111111214441224
4. 221111111133333333333333322211
5. 112111111111433332222221111111

Kod četvrtog klastera nije toliko jasno izraženo pripadnost jednog regiona, mada se može primetiti da dominiraju regioni 2 i 3. Kako postoje 5 regiona, a izabrali smo 4 klastera, ovo ima neke logike da ih je algoritam sam spojio kao jedan region. Slede primeri 5 nasumičnih sekvenci iz četvrtog klastera:

1. 22222222222223333333333333222
2. 44333222223334443334422222222
3. 42332222344444323344444112222
4. 22222333333322224444332222222
5. 22222212334444444444444333222

5. Zaključak:

Klasterizacija podataka je težak zadatak, a postaje još teži kada se doda vremenska dimenzija. Uopšte sam način zapisivanja i predstavljanja vremenske sekvence koja može da traje neograničeno na način da se iz nje mogu izvući podaci i bude pogodna za mašinsku obradu nije lak zadatak. U ovom radu je izabrano skaliranje na sekvencu određene dužine čime se izgubio deo podataka i ovo predstavlja jedan od delova gde mogu da se naprave poboljšanja. Kao neki vid daljeg unapređivanja sledio bi pokušaj obučavanja autoenkodera na sekvencama varijabilne dužine.

Pored toga trebalo bi obratiti pažnju i na sastavljanje testova gde bi regije trebale da imaju minimalnu veličinu i na njihovo pozicioniranje na ekranu kako bi merni uređaj bolje radi i dobili se kvalitetniji podaci.

Autoenkoder se pokazao kao validna strategija u smislu da može da pronađe skrivene osobine sekvenci i skrati njihovu reprezentaciju. Samo te osobine nisu uvek nama razumljive. Zanimljivo bi bilo videti i mapirati na šta se neuroni okidaju odnosno šta ih aktivira. Ovo bi dalo uvid koji bi kasnije mogao poslužiti u klasterizaciji za bolje razumevanje i pri izboru broja klastera. Trenutno u projektu su izabrana 4 klastera jer za njih je bilo jasno šta izdvajaju, i po čemu biraju celine. Tokom izrade projekta rađena su i aglomerativna klasterovanje sa više klastera i rezultati su pratili neku logiku, samo nije bilo najjasnije šta je tačno uticalo da nešto uđe u taj klaster (dešavalo se da sekvence u nekom klasteru počinju određenim regionom ali mogle su se pronaći isto tako sekvence koje nisu u tom klasteru a počinju sa tim istim regionom, te nije bilo jasno da li je to početak sekvence tim regionom faktor koji određuje pripadnost tom klasteru)

Reference:

[1] Jeffery, L.; Rhodes, G. (2011). "Insights into the development of face recognition mechanisms revealed by face after effects". *British Journal of Psychology*.

[2] Sepp Hochreiter; Jurgen Schmidhuber (1997) "LONG SHORT-TERM MEMORY"

[3] Jitendra Kumara; Rimsha Goomerb; Ashutosh Kumar Singh (2017) "Long Short Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model For Cloud Datacenters"

[4] Gabor Petnehazi (2019) "Recurrent Neural Networks for Time Series Forecasting"

[5] D. E. Rumelhart; G. E. Hinton; R. J. Williams; "Parallel distributed processing: Explorations in the microstructure of cognition"

[6] G. Hinton; R. Salakhutdinov; (2006) "Reducing the dimensionality of data with neural networks"

[7] D. Erhan; Y. Bengio; A. Courville; P.-A. Manzagol; P. Vincent; S. Bengio (2010) "Why does unsupervised pre-training help deep learning?"

[8] Wenjie Peia; David M.J. Tax (2018) "Unsupervised Learning of Sequence Representations by Autoencoders "